修 士 論 文 概 要 書

CD

2010 年 2 月提出　　学籍番号 5108B046–9

専門分野	情報理工学専攻	氏 名	小林 優太	指 導 教 員	大附 辰夫 印
研究指導	情報アーキテクチャ研究				
研 究 題 目	組み込みシステムにおける 2 階層ユニファイドキャッシュを対象とした キャッシュ構成最適化手法に関する研究				

1 はじめに

近年の CPU 等の処理装置の高速化により，メモリ等の記憶装置との性能差拡大が問題となっている．これを解決する 1 つの手法がキャッシュである．しかし，対象となるアプリケーションによっては，キャッシュ容量が小さすぎるなどキャッシュ構成が適合せず，キャッシュミスが頻繁に発生する場合がある．組み込みシステムは汎用プロセッサとは違い，実行するアプリケーションが決定しているため，特定アプリケーションのみ処理できれば良い．よって，対象となるアプリケーションごとにキャッシュ構成を決定することは，実行速度，面積，消費エネルギーなどの面で有効な手段となるため，キャッシュ構成の最適化手法が必要となる．

キャッシュ構成最適化手法の 1 つに，メモリ参照履歴であるトレースデータから各キャッシュ構成のキャッシュヒット/ミス数をシミュレーションする手法がある．このキャッシュヒット/ミス数を基に評価式を導入して，メモリアクセス時間最小やメモリ消費エネルギー最小となるキャッシュ構成を探索する．しかし，単純に全てのキャッシュ構成においてシミュレーションを実行した場合，膨大なキャッシュ構成に対してシミュレーションが必要となるため，シミュレーション時間の増大が問題となる．また，既存手法として 2 階層ハーバードキャッシュを対象とした手法は多く提案されているが，一般的な 2 階層ユニファイドキャッシュを対象として，必ず最適解を探索する手法は提案されていない．

こうした背景から，本論文では我々の研究室が提案している 2 階層ハーバードキャッシュを対象とした CRCB 手法を 2 階層ユニファイドキャッシュに適用し，さらなる高速化手法である CRCB-U1 (CRCB for L2 Unified cache) 手法，CRCB-U2 手法を提案する．提案手法は対象とする全キャッシュ構成に対してヒット/ミス数を取得することができ，必ず最適なキャッシュ構成を選択することができる．

2 キャッシュシミュレーション高速化手法

キャッシュ構成のパラメータとして，セット数 s，ブロックサイズ b，連想度 a，容量 t があり，$t = s \times b \times a$ が成り立つ．L1 データキャッシュのセット数/ブロックサイズ/連想度をそれぞれ s_d，b_d，a_d，L1 命令キャッシュのセット数/ブロックサイズ/連想度をそれぞれ s_i，b_i，a_i，L2 ユニファイドキャッシュのセット数/ブロックサイズ/連想度をそれぞれ s_{L2}，b_{L2}，a_{L2} とするとき，あるキャッシュ構成 c を $c = ((s_d, b_d, a_d), (s_i, b_i, a_i), (s_{L2}, b_{L2}, a_{L2}))$ のように表す．本論文ではアプリケーションのトレースデータから，この 9 つのパラメータを変化させたときの全キャッシュ構成におけるヒット/ミス数を取得することを目的とする．

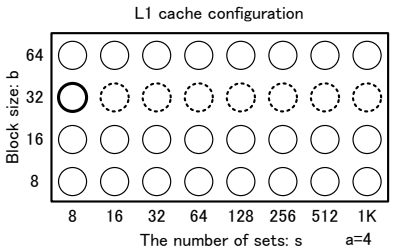

図 1: CRCB1 手法よるシミュレーション省略例．$s = 8$, $b = 32$ の 1 番目のブロックで tag がヒットした場合，$8 < s$, $b = 32$ のヒットミスを省略できる．

2.1 2 階層ハーバードキャッシュ

Mattoson によると，2 つのキャッシュ構成 c_1, c_2 について，キャッシュ構成 c_1 の要素が全てキャッシュ構成 c_2 の要素に含まれていれば $c_1 \subset c_2$ と書き，キャッシュ構成 c_2 でキャッシュミスが起きればキャッシュ構成 c_1 でもキャッシュミスが起き，キャッシュ構成 c_1 でキャッシュヒットすればキャッシュ構成 c_2 でもキャッシュヒットすることが示されている (Inclusion Property)．我々は，Janapsatya らが提案する複数連想度同時探索手法と組み合わせて，2 階層ハーバードキャッシュを対象とした CRCB1 手法，CRCB2 手法，CRCB-T1 手法，CRCB-T2 手法を提案している．各手法は，Inclusion Property をそれぞれ以下のキャッシュの要素に適用している．

- 複数連想度同時探索：Inclusion Property を連想度に適用
- CRCB1：Inclusion Property をセット数に適用
- CRCB2：Inclusion Property をブロックサイズに適用
- CRCB-T1：Inclusion Property を 2 階層キャッシュに適用
- CRCB-T2：Inclusion Property を初期参照ミスに適用

図 1 に CRCB1 手法成立時のシミュレーションの省略例を示す．この CRCB 手法により，命令キャッシュを対象としたキャッシュ構成 $c = (-, (s_i, b_i, a_i), (s_{L2}, b_{L2}, a_{L2}))$，または，データキャッシュを対象としたキャッシュ構成 $c = ((s_d, b_d, a_d), -, (s_{L2}, b_{L2}, a_{L2}))$ の全キャッシュ構成のヒット/ミス数を高速に取得することが可能となる．

2.2 2 階層ユニファイドキャッシュ

CRCB 手法を 2 階層ユニファイドキャッシュシミュレーションに拡張する手法を提案する．CRCB 手法の拡張には L1 キャッシュのどちらか一方の構成を固定することを考える (以降の説明ではデータキャッシュを固定するものとする)．これにより，L1 データキャッシュを (s_d^0, b_d^0, a_d^0) と固定した時のキャッシュ構成 $((s_d^0, b_d^0, a_d^0), (s_i, b_i, a_i), (s_{L2}, b_{L2}, a_{L2}))$ の **L1 データキャッシュにおけるミスアクセス**は $(-, (s_i, b_i, a_i), (s_{L2}, b_{L2}, a_{L2}))$ の **L1 命令キャッシュ構成において必ずミスするアクセス**としてシミュレーションしていると見なすこともでき，擬似的に命令キャッシュを対象とした 2 階層ハーバードキャッ

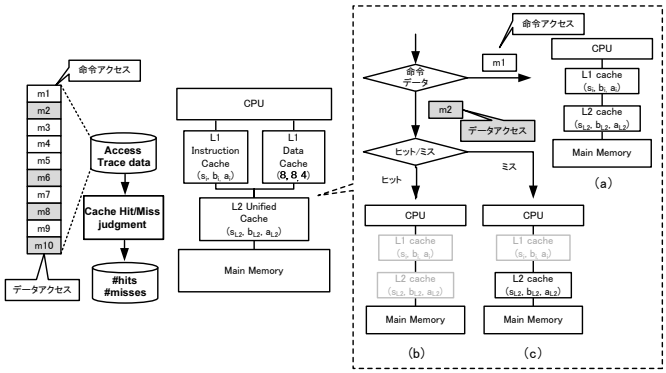

図 2: L1 データキャッシュ構成固定による 2 階層ユニファイドキャッシュシミュレーション.

シュシミュレーションと見なすことができる（図 2）. 以上のことから，固定する L1 データキャッシュを変化させながら CRCB 手法のアルゴリズムを繰り返すことで 2 階層ユニファイドキャッシュに対しても全キャッシュ構成のヒット/ミス数を取得することができる（ただし，CRCB-T1 手法は適用できない）.

2.2.1 CRCB-U1 手法

CRCB-U1 手法は 2 階層ユニファイドキャッシュの初期参照ミスに Inclusion Property を適用した高速化手法である. キャッシュのブロックサイズを変化させなければ初期参照ミス数は変化しないという性質などから，固定した L1 データキャッシュ構成のブロックサイズのみを増大させていったときに総ミス数が初期参照ミス数と同数になると，それ以降ブロックサイズを増大させてもキャッシュミスの振る舞いは完全に同一になる. これにより，図のように総ミス数が初期参照ミス数と等しくなると (太線の円)，それ以降のキャッシュ構成 (点線の円) のシミュレーションを省略できる.

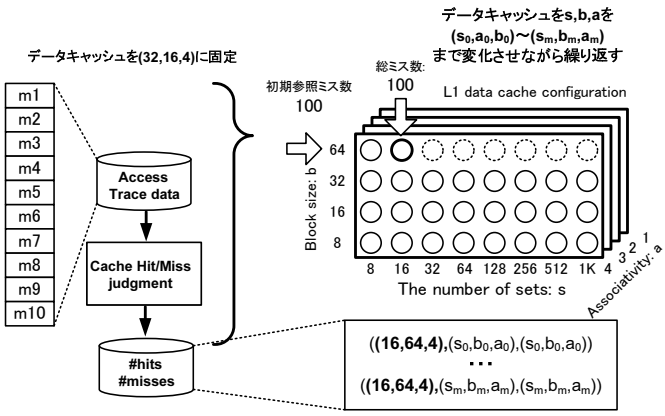

図 3: CRCB-U1 手法

2.2.2 CRCB-U2 手法

2 階層ハーバードキャッシュを対象とした CRCB 手法は，対象となる全キャッシュ構成に対して 1 メモリアクセスごとに並列にシミュレーションする. そのため，Inclusion Property から必ずヒット/ミスする構成のシミュレーションを省略することができる. しかし，2 階層ユニファイドキャッシュでは，L1 データキャッシュを 1 構成に固定してシミュレーションするため，Inclusion Property の性質は成立しているが，シミュレーションの省略はできない. そこで，CRCB1 手法，CRCB2 手法が成立した時に，あるキャッシュ構成のシミュレーションを省略する代わりに，対応するトレースデータのメモリアクセスを以降のシミュレーションから除外することでシミュレーションの高速化をする.

3 計算機実験

提案するキャッシュ構成最適化システムを C++言語によって計算機上に実装した. 実験環境は，OS が Debian GNU/Linux，CPU は Intel Xeon 3.40GHz，メモリ容量 4GB，対象アプリケーションは MediaBench を利用している. 提案システムは総メモリアクセス時間最小または総メモリ消費エネルギー最小となるキャッシュ構成を探索する. 図 4 に総メモリアクセス時間最小としたときのシミュレーション時間を示す（全探索シミュレーションを 1 とする）. 図の CRCB は複数連想度同時探索手法，CRCB1 手法，CRCB2 手法，CRCB-T2 手法をユニファイドキャッシュシミュレーションに拡張した手法である. また，CRCB+U1，CRCB+U1+U2 は，それぞれ CRCB-U1 手法，CRCB-U2 手法を適用したものである. 各表示中のアプリケーション項目は (E) がエンコード，(D) がデコードを表している. また，全探索見積値は，キャッシュ1 構成のシミュレーションを実行し，(キャッシュ 1 構成のシミュレーション時間) × (全キャッシュ構成数：5782690) で計算している.

CRCB+U1+U2 は全探索手法の見積値と比較して最大 3761.08 倍，最小 109.77 倍，平均 963.81 倍高速化されている. また，CRCU+U1+U2 は CRCB と比較して最大 2.69 倍，最小で 1.08 倍，平均 1.90 倍高速化している. 全探索見積値では，1709 日掛かる G721 エンコードのシミュレーションが，提案手法により約 7 日でシミュレーションが完了するため，提案手法は有効であると言える.

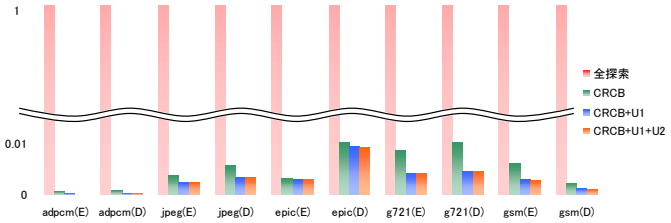

図 4: 実験結果（総メモリアクセス時間最小）

4 おわりに

本論文では，2 階層ユニファイドキャッシュを対象としたキャッシュ構成最適化手法を提案した. 提案手法は，キャッシュの性質である Inclusion Property を利用することで，シミュレーションの高速化を実現しつつ全キャッシュ構成のヒット/ミス数を正確にシミュレーションすることを可能とした. 今後の課題としては，トレースデータの読み込み回数削減による高速化手法の提案やマルチコアへの対応が考えられる.

研究業績　電子情報通信学会 VLD 研究会 2 件